Algorithms – CS-27200/37000   Homework – February 11, 2005

Instructor: László Babai   Ry-164   e-mail: laci@cs.uchicago.edu

TA SCHEDULE: TA sessions are held in Ryerson-255, Tuesday and Thursday 5–6pm, Saturday 11am–noon, and (this is new) **Wednesday after class** 12:30–1:20 or 1:30–2:20 depending on demand. Indicate your interest in the Wednesday session to the instructor immediately after class. (The Wednesday evening sessions are discontinued.)

ADVICE. Take advantage of the TA sessions.

Check the class website, http://www.classes.cs.uchicago.edu/current/27200-1.

HOMEWORK. Please **print your name on each sheet.** Print "U" next to your name if you seek 27200 credit and "G" if you seek 37000 credit. Undergraduates receive the stated number of points as *bonus points* for "G only" problems. – Please try to make your solutions readable. Unless expressly stated otherwise, all solutions are due at the **beginning of the next class.**

COMING SOON: applications of number theory. **Review** related material from Discrete Mathematics: greatest common divisor, Euclid's algorithm, congruences, Fermat's Little Theorem, Chinese Remainder Theorem.

DATES TO REMEMBER: Mon Feb 21: Quiz 2,   Mon Mar 7: Midterm 2,   Fri Mar 11: Last class. ATTENDANCE REQUIRED.   Fri Mar 18, 10:30–12:30: Final Exam

12.1 (U,G) (4+5+5+5 points) Recall that the Fibonacci numbers F_n are defined by the recurrence $F_n = F_{n-1} + F_{n-2}$ with initial values $F_0 = 0$ and $F_1 = 1$.

The *height* $h(x)$ of a node x in a rooted tree is the length of the longest path leading from x to a leaf. (Leaves have height zero.) The *leftheight* $lh(x)$ of a node in a binary tree is $1 + h(L(x))$ where $L(x)$ denotes the left child of x. If x has no left child ($L(x)$ =NIL) then $lh(x) = 0$. The *rightheight* $rh(x)$ is defined similarly.

An *AVL-tree* is a binary tree in which every node x satisfies the inequality $|\,lh(x) - rh(x)| \le 1$. (The tree is "almost balanced" at every node.)

Let $m(t)$ denote the minimum number of nodes of an AVL-tree of height t. (The height of a rooted tree is the height of its root.) So $m(0) = 1, m(1) = 2, m(2) = 4$.

(a) List $m(t)$ for $t \le 5$ and draw the corresponding minimal AVL-trees.

(b) Find a simple recurrence satisfied by the sequence $m(t)$. Prove your answer.

(c) Determine the value $m(t)$ for all t. Your answer should be a very simple expression in terms of Fibonacci numbers. Prove your answer.

(d) Prove: if an AVL-tree has n nodes and height h then $h \lesssim c \log n$ where $c = 1/\log g$ where $g = (1+\sqrt{5})/2$ is the golden ratio.

12.2 (U,G) (15 points) Modify the AVL-tree data structure to support the RANGESUM(α, β) request (in addition to the requests it already supports: FIND(key), INSERT(key), DELETE(node)).

On input (α, β) (reals), RANGESUM(α, β) must output the sum of all those currently stored keys x satisfying $\alpha \leq x < \beta$. Each request, including RANGESUM, should be served in $O(\log n)$ steps, where n is the number of currently stored keys. (One step is an arithmetic operation, comparison, or bookkeeping (such as copying a link)).

Concentrate on the *additional information* you need to store and maintain at each node. Give an accurate description of this information and its location, and state how it is maintained under INSERT without rotation. (Ignore rotations and DELETE). Finally, describe and *prove* that this additional information allows RANGESUM requests to be served in $O(\log n)$ steps.